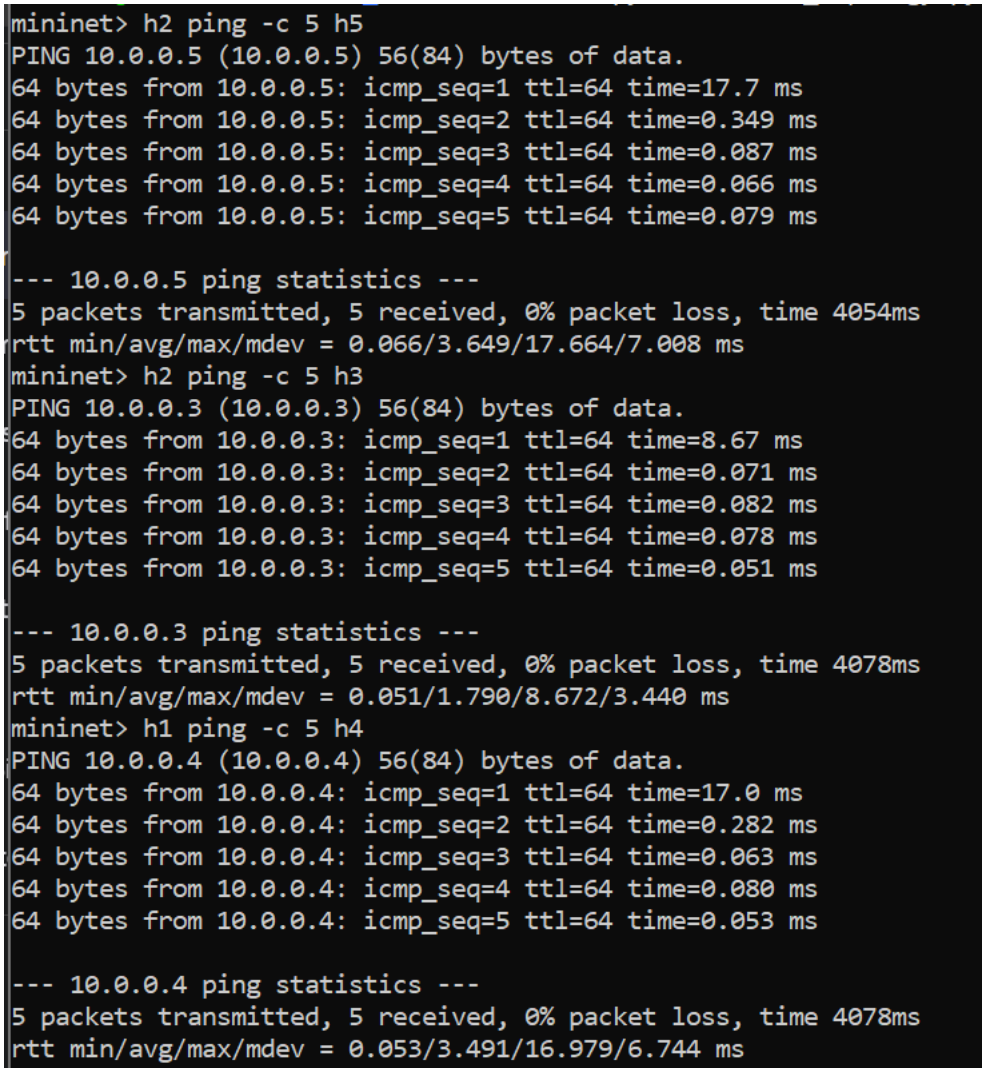
Programming Assignment 3: SDN and Ryu

(jcs232542- SHARATH KUMAR REDDY GANDHAM)

Part 1:

For Learning_switch:



For Controller_hub:

```
mininet> h2 ping -c 5 h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=18.1 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=9.78 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=9.32 ms
64 bytes from 10.0.0.5: icmp_seq=4 ttl=64 time=8.83 ms
64 bytes from 10.0.0.5: icmp_seq=5 ttl=64 time=9.75 ms

--- 10.0.0.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 8.829/11.162/18.137/3.504 ms
mininet> h2 ping -c 5 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=12.3 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=8.77 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=4.97 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=5.52 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=6.47 ms

--- 10.0.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 4.967/7.602/12.286/2.677 ms
mininet> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=13.9 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=7.82 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=6.14 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=7.81 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=9.63 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 6.142/9.063/13.916/2.665 ms
```

Here we can observe that the latencies for the learning switch are much smaller than the controller hub because the controller hub needs to use the controller to find the way to send the received packet, which overheads the hub to send the packet. Here it does not store anything about the hosts.

Whereas in the learning switch(it uses layer2 mechanism) at every first-time learning switch also does the same thing done by the hub, but for the next subsequent packets of the same destination it just sends the packet directly without any delay or overhead because here it stores the previous path to send.

Throughput test:

Learning Switch:

```
mininet> iperf h1 h5
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['33.9 Gbits/sec', '34.0 Gbits/sec']
```

Controller Hub:

```
mininet> iperf h1 h5
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['6.20 Mbits/sec', '7.26 Mbits/sec']
```

The Learning Switch offers higher local network performance due to its efficient, decentralized operation, while the Controller Hub introduces lower performance but provides more centralized control

After ping all the rules installed:

Learning_switch:

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=38.143s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=38.141s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=38.128s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=38.126s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=38.112s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=38.110s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=38.094s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=38.091s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=38.076s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=38.070s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=38.054s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=38.048s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=38.031s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=38.027s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=38.003s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=37.999s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=37.980s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=37.975s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=42.765s, table=0, n_packets=28, n_bytes=1680, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=38.119s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=38.113s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=38.100s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=38.094s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=38.062s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=38.050s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=38.040s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=38.030s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=38.009s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=38s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=37.989s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=37.977s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=37.952s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=37.950s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=42.758s, table=0, n_packets=24, n_bytes=1400, priority=0 actions=CONTROLLER:65535
```

Controller_hub:

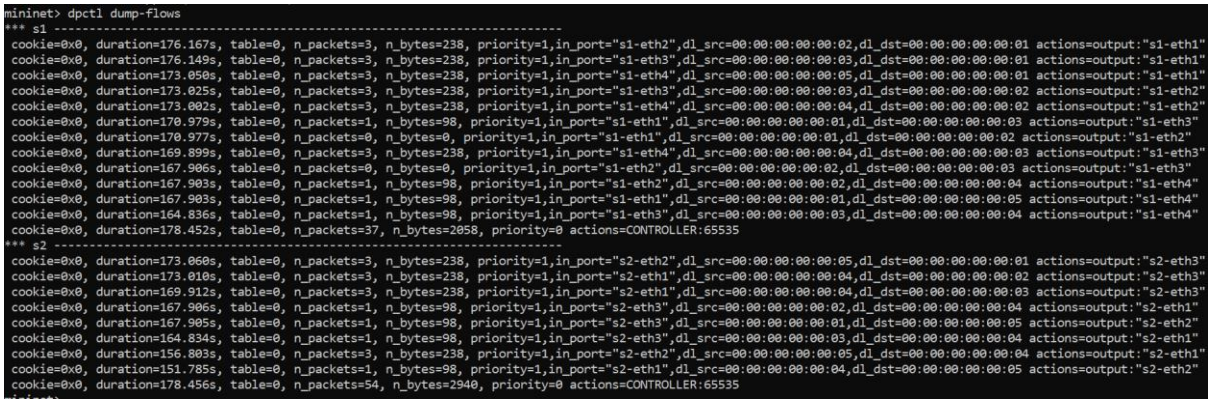
```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=7.583s, table=0, n_packets=60, n_bytes=4760, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=7.587s, table=0, n_packets=60, n_bytes=4760, priority=0 actions=CONTROLLER:65535
mininet>
```

part-2:

Firewall and moniter:

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X h5
h2 -> h1 h3 h4 X
h3 -> h1 h2 h4 X
h4 -> X h2 h3 h5
h5 -> h1 X X h4
*** Results: 30% dropped (14/20 received)
```

Installed Rules in both switches:



```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=176.167s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=176.149s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=173.050s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=173.025s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=173.002s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=170.979s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=170.977s, table=0, n_packets=0, n_bytes=0, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=169.899s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=167.906s, table=0, n_packets=0, n_bytes=0, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=167.903s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=167.903s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=164.836s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=178.452s, table=0, n_packets=37, n_bytes=2058, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=173.060s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=173.010s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=169.912s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=167.906s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=167.905s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=164.834s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=156.803s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=151.785s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=178.456s, table=0, n_packets=54, n_bytes=2940, priority=0 actions=CONTROLLER:65535
```

Q)Can you think of ways to minimize the number of firewall rules on the switch?

By the following way we can minimize the number of rules in firewall:

Wildcard Rules: Instead of having specific rules for each pair of IP addresses that you want to block, consider having wildcard rules that cover multiple IP addresses. For example, you can create rules to block entire subnets or address ranges. This reduces the number of rules required.

Rate-Limiting: Instead of blocking traffic completely, you can use rate-limiting to restrict the amount of traffic that can flow between certain IP address pairs. This way, you can avoid creating extensive block rules.

Consolidate Rules: If you have multiple IP address pairs to block that share common characteristics or patterns, you can consolidate them into a single rule. This can be achieved by specifying multiple source or destination addresses in a single rule.

Blacklists and Whitelists: Consider maintaining a dynamic blacklist and whitelist of IP addresses. Instead of static rules, update these lists in real-time based on network activity and policies.

Use Layer 4 Rules: If your switch supports layer 4 (transport layer) rules, you can create rules based on ports and protocols, which can be more concise than IP address-based rules.

Prioritize Rules: If you have a large number of rules, prioritize them based on the importance of the traffic. Higher-priority rules can be checked first, and lower-priority rules only if the higher-priority rules don't match.

Q) Suppose the network operator intends to implement firewall policies in real time. How can she ensure that the pre-existing rules do not interfere with the firewall policy?

Document Existing Rules:

Carefully document and analyze the pre-existing firewall rules to understand their purpose and impact on network traffic.

Review and Audit:

Conduct a thorough review and audit of the existing firewall rules to identify any unnecessary or conflicting rules. Remove or update any obsolete rules.

Use Rule Ordering:

Firewalls typically process rules in a specific order. Ensure that your new rules are placed appropriately in the rule order. Most firewalls use a top-down rule evaluation approach, meaning the first matching rule is applied. Make sure that your new rules do not get overshadowed by more general rules placed earlier.

Prioritize Rules:

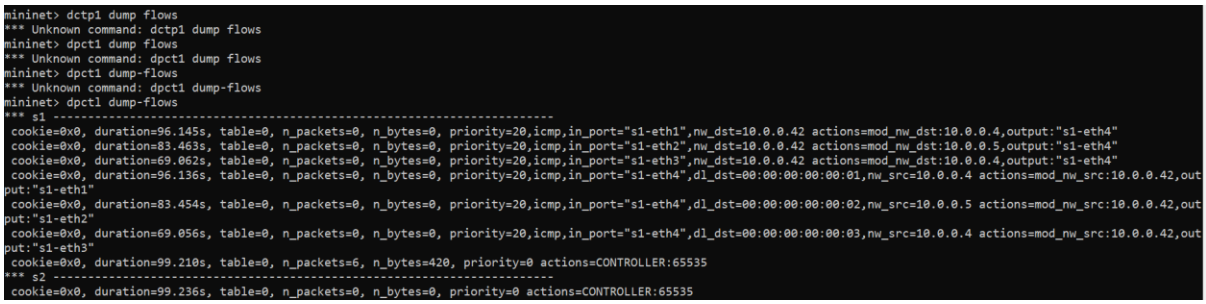
Prioritize your firewall rules. Critical or security-sensitive rules should be placed higher in the rule order. Rules that are less critical can be placed lower in the order.

Part-3:

Q) Have the three hosts (H1, H2, and H3) ping the virtual IP and report the installed rule on the switches.

Hosts H1, H2, and H3 can ping the virtual IP address 10.0.0.42 as if it were a regular server. The load balancer on S1 will distribute the traffic to H4 and H5 in a round-robin manner.

Installed rules:



Q) If you were to implement a load balancing policy that considers the load on these servers, what additional steps would you take?

For implementing a more sophisticated load balancing policy that considers the load on the servers, you would need to implement the following additional steps:

- You would need to periodically check the status and performance of H4 and H5 using techniques like health checks and monitoring the server's resource utilization (e.g., CPU, memory, and network).
- Implement dynamic load balancing: Modify the load balancer rules dynamically based on server load. For example, you can send more traffic to the server with a lower load and less traffic to the server with a higher load. You can use tools like dynamic weighted round-robin or least connections algorithms.
- Handle server failover: Implement logic to handle server failures and redirect traffic to healthy servers.
- Consider session persistence: If your application requires session persistence (where all requests from a specific client go to the same server), you would need to implement additional logic to achieve that, such as using cookies or source IP-based persistence.